AD NUMBER

AD881171

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

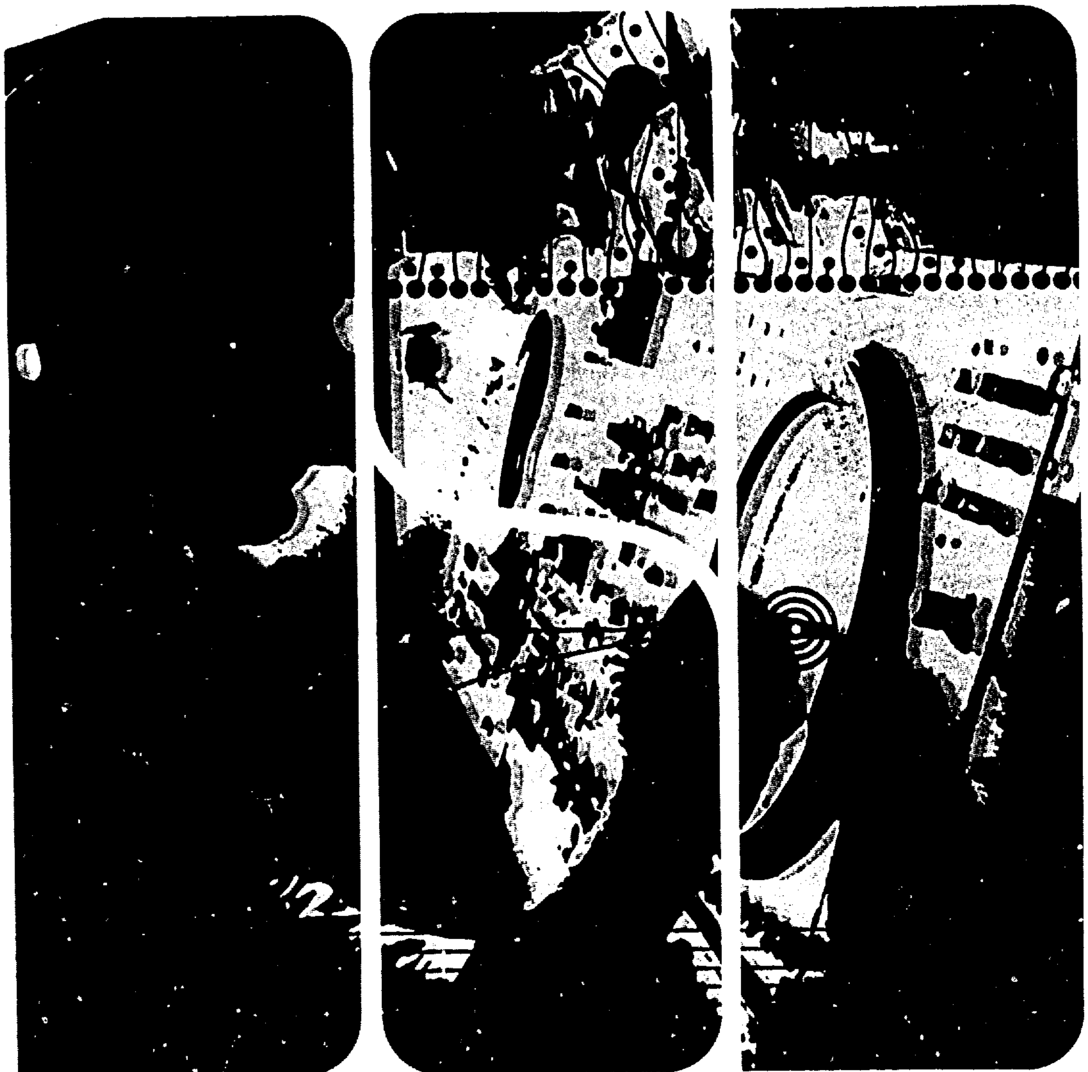
Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; 1 Mar 1971. Other requests shall be referred to Attn: SMUFA-L3300, U. S. Army, Frankford Arsenal, Bridge and Tacony Streets Philadelphia, PA 19137.

AUTHORITY

FA, D/A ltr, 9 Oct 1974

(3)

Data Systems Division · Litton Industries



MS 00834

'NAL REPORT
A PARAMETRIC STUDY OF THE ADVANCED FORWARD AREA AIR DEFENSE WEAPON SYSTEM (AFAADS)
VOLUME II – SIMULATION

2 October 1970

Submitted to

U. S. Army, Frankford Arsenal
Bridge and Tacony Streets
Philadelphia, Pennsylvania 19137

Prepared by:

Data Systems Division
Litton Systems, Inc.
8000 Woodley Avenue
Van Nuys, California 91409

FOREWORD

This report describes the research effort of the Data Systems Division of Litton Industries, Inc., under Contract No. DAAG05-70-C-0328, with the U.S. Department of the Army, Frankford Arsenal. The objective was to perform a parametric study of the Advanced Forward Area Air Defense (AFAADS) Weapon System.

The original RFP, to which this contractual effort is responsive, was titled 'Parametric Study, Advanced Forward Area Air Defense Weapons System (AFAADS),' and called for studies specific to an antiaircraft gun system based on a weapon having the same general characteristics as the 37mm Gatling Gun.

In the intervening time, since the 3 May 1968 date of issue of the RFP, the Army generalized the use of the acronym 'AFAADS' to include all elements of the forward area air defense of the Army, including missiles. At the present time, it appears that the effort contracted for could more properly be called 'LOFADS,' Low Altitude Forward Area Air Defense.

However, in order to maintain consistency with the wording of the contract, the term AFAADS has been used uniformly in the present report. The reader should remain aware of the fact that we are dealing here only with the antiaircraft gun system, which in turn is an element of the complete system for the forward air defense of the Army.

The report is included in two volumes. Volume I comprises the analyses and conclusions derived from an in-depth study by Mr. Herbert K. Weiss. Also provided therein are descriptions of the AFAADS System Concept as well as characteristics of AFAADS and its environment.

Volume II describes a computer simulation model used to provide the capability of predicting the performance of proposed (or existing) antiaircraft gun systems against an aircraft in a one-to-one combat situation. Results of this effort ~~by Mr. Martin P. Ginsberg~~ were used to derive recommended configurations and conclusions as presented in Volume I.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1
INTRODUCTION

The purpose of the AFAADS computer-simulation model is to provide the capability of predicting the performance of proposed (or existing) antiaircraft gun systems against an aircraft in a one-to-one combat situation.

The model is not particular to one gun system or aircraft but has the ability to model a variety of systems. The modular design of the simulation allows each capability to be progressively refined as more detailed definition of an element of the system is developed. The simulator at present includes the facilities to:

a. Model realistic aircraft paths.

b. Model sensor behavior including the effects of:

   (1) Sensor lag.

   (2) Regenerative tracking.

   (3) Time-correlated sensor noise and aim wander.

   (4) Data smoothing.

c. Model the following prediction methods:

   (1) Linear.

   (2) Linear plus energy corrections.

   (3) Quadratic.

   (4) Quadratic plus energy corrections.

   (5) Defense of a known point.

   (6) Simple polar.

d. Model the ballistics of any shell with trajectory characteristics suitable to representation as a solution of the two parameter differential equation:

$$dV = -k\, V^{\alpha}\, dt \tag{1.1}$$

e. Couple the selected prediction algorithm and the selected ballistics in the time of flight and shell effectiveness computations.

f. Model gun motion including the effects of:

   (1) Gun lag.

   (2) Regenerative tracking.

g. Compute miss distance at the time of closest approach of the shell to the aircraft.

h. Generate periodic and/or summary reports of the output from the:

   (1) Sensor.

   (2) Data smoother.

   (3) Predictor.

   (4) Miss-distance calculation.

i. Generate a summary report including the following items in any combination:

   (1) The single shot kill probability of shots equally spaced in time over the duration of the aircraft course.

   (2) The expected number of killing hits for each burst of a series of equally spaced disjoint bursts over the duration of the aircraft course.

   (3) The probability that the aircraft is 'killed' for each burst of a series of equally spaced disjoint bursts over the duration of the aircraft course.

   (4) The number of bursts of a series of equally-spaced disjoint bursts over the duration of the aircraft course which have kill probabilities in excess of each of five reference probabilities.

   (5) The number of bursts of a series equally-spaced disjoint bursts over the duration of the aircraft course which have miss distances not exceeding each of five reference distances.

   (6) The integral of the single-shot kill probability over the duration of the aircraft course.

The computer program is written in basic FORTRAN IV and is currently being executed on the RAX time-sharing system. No timing estimates are available as to the actual running time, however, it is less than 20 seconds of RAX time for a typical problem. Actual 360/40 central processor time is probably much less than the RAX terminal time.

The model is structured in a manner which maintains the 'real world' information pattern of the simulated gun system. This is accomplished by restricting the access of each simulated gun component to only that information which would be available in the 'real world.'

The manner in which information is acquired, generated, and transmitted through the system is presented in Figure 1-1. The flow is as follows: from aircraft position to the Data Acquisition Event, which fills the SENSED POSITION File; to the Prediction Event, which accesses the SENSED POSITION File and fills the PREDICTED POSITION File; to the Gun

Motion and Firing Event, which accesses the PREDICTED POSITION File and fills the SHELL POSITION File.

The preceding information flow pattern demonstrates an important feature of the model. Note that, as in the 'real world,' any action which the gun system takes (or can take) is ultimately based on information acquired through its sensing system.

In the following sections the model capability is described in more detail. To facilitate the discussion, the model is viewed in four parts; three sets of functionally related program modules and a communication pool through which information is transmitted between the various modules.

The first set of modules is used to control the simulation process. It is composed of the main program and one subroutine which maintains a schedule of events and provides for central time accounting.

The second set of modules is composed of the Gun System Functional Event Routines. This is the set of routines which model the various components of the gun system under study. There are four functional event routines:

a. Sensor data acquisition.

b. Regenerative tracking switch.

c. Prediction.

d. Gun motion and firing.

The third set of modules is the utility package. These modules are used by the event routines and/or each other and may not be invoked by the system scheduler. Report routines are excepted from this rule since they do not change the contents of any communication pool variables. Included in this set are:

a. Aircraft positioner.

b. Time of flight.

c. Prediction algorithm.

d. Random number generator.

e. Coordinate transformation package.

f. Report generators.

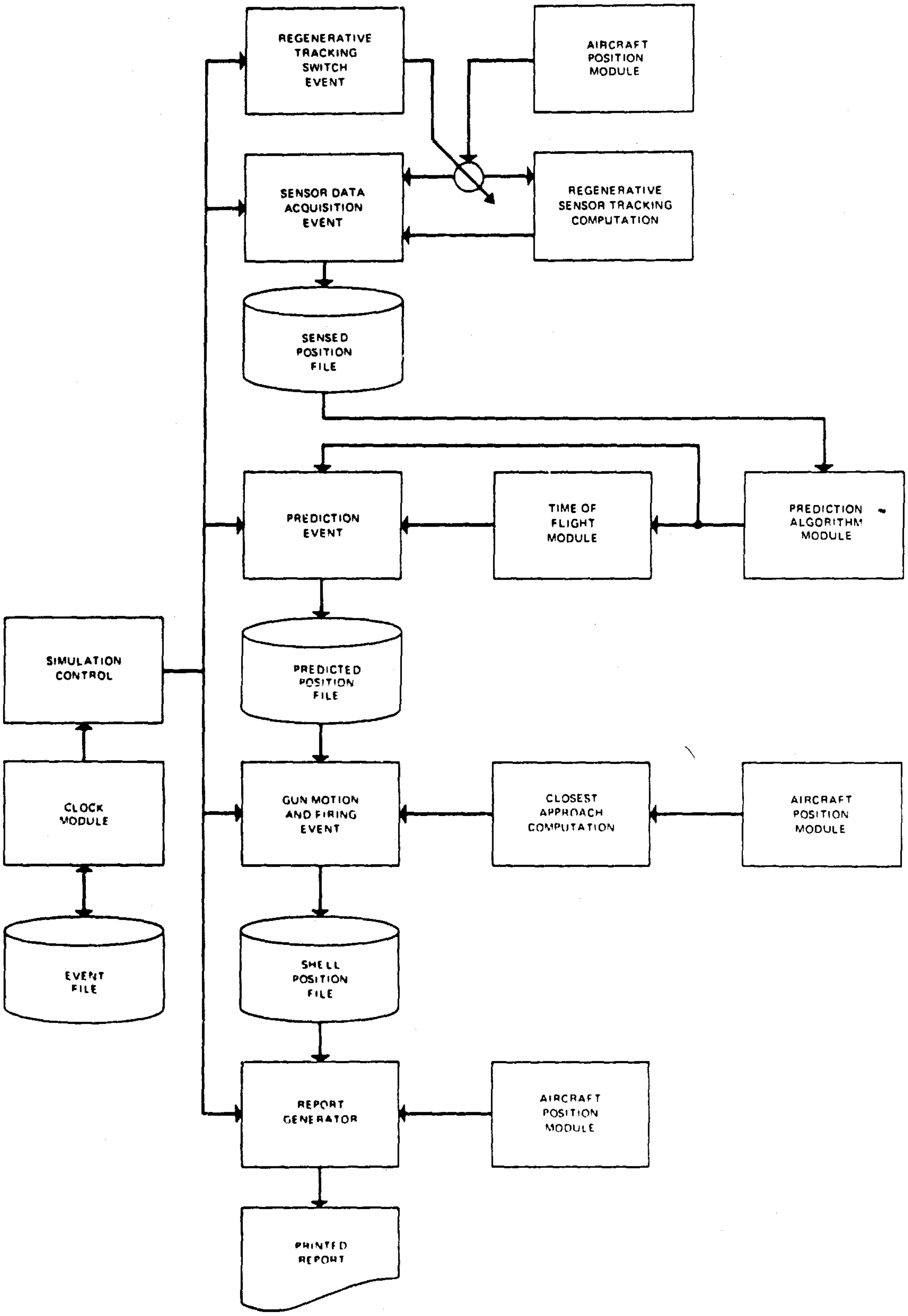


Figure 1-1. Information Flow Through the AFAADS Simulation Model

SECTION 2
SIMULATION CONTROL

The simulation model represents the combat situation under study as a series of events centered around the antiaircraft gun system. Event sequencing and initialization tasks are performed by the master control program and the clock module. The initialization phase consists of reading input parameters, performing and storing the results of once-only computations, and scheduling the first (a sensor data acquisition event) and last (a report event) events to be processed. Initialization tasks are accomplished by special modules which are called by the master control program prior to entering the event sequencing mode.

The logic of the control process is presented in Figure 2-1. The particular order of events, which will occur, is determined by the gun system and the aircraft path under study. There are, however, some principles worth noting: for example, the first event to occur will always be Event Type 1, and it will continue to occur at its specified period (input during initialization phase) until the end simulation event. With the exception of the end simulation event, no other events will occur unless they are explicitly scheduled by Event Type 1 or some other module, either directly or indirectly scheduled by an occurrence of Event Type 1. The occurrence of Event Type 2 indicates the end of the current sequence of simulation events.

One of two actions are initiated after the occurrence of Event Type 2. If there are more replications to be performed: the event file is cleared, all flags are turned off, the initial Event Type 1 and end simulation events are scheduled, and the simulation process continues. If no more replications are required, the summary statistics over replications are printed, and the simulation process is terminated.

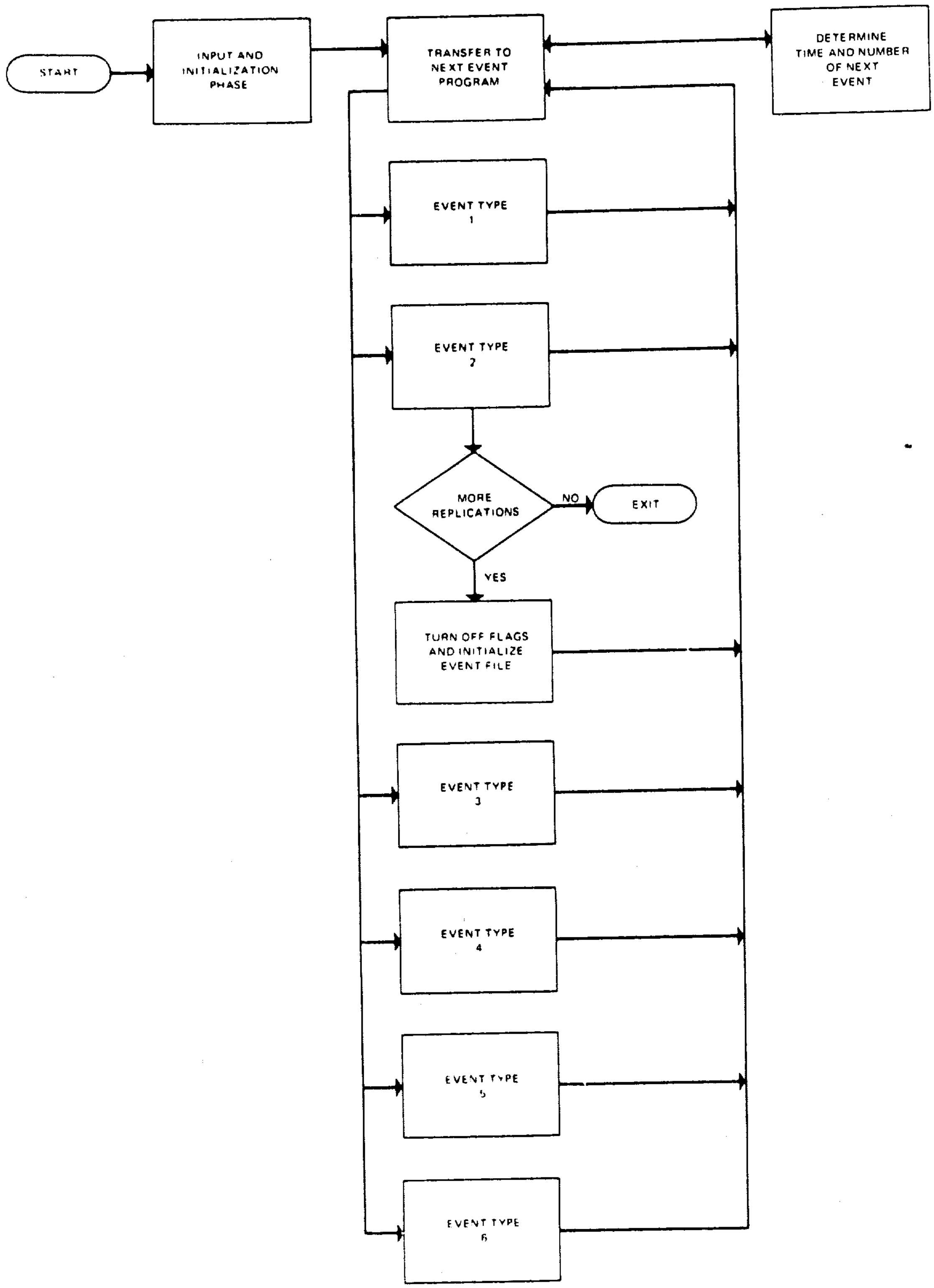


Figure 2-1. Logic Diagram of the Control Process

SECTION 3
EVENTS

3.1 SENSOR DATA ACQUISITION EVENT

In the real world all gun system functions, which require a knowledge of aircraft position and/or rate parameters, acquire the information either directly or indirectly from the gun sensor. In the computer model the sensor function is modeled by the Sensor Data Acquisition Event. On each entry, this event routine stores current values of sensed aircraft position in the SENSED DATA File. The gun-system's data smoother is also modeled in this routine and its outputs, smoothed position and rate values, are also placed in the data file.

In the following discussion, upper case letters will be used for true values, primed upper case letters for sensed values, underscored upper case letters for smoothed values of position. and dot notation to indicate rates and accelerations. The time to which values pertain will be indicated by subscript expressed relative to current time. Thus, if A is a position coordinate:

A_0 = Current true value.

A'_0 = Current sensed value.

$\underline{A}_0$ = Current smoothed value.

$\dot{\underline{A}}_0$ = Current smoothed rate value.

$A'_{-\Delta t}$ = Sensed value at current time minus Δt.

The six dimensional transformations from cartesian to polar, and from polar to cartesian coordinates, will be represented by the operators; 'CTOP' and 'PTOC.' Thus:

$$(R, \theta, \phi, \dot{R}, \dot{\theta}, \dot{\phi}) = \text{CTOP}\,[X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}] \qquad (3.1)$$

and

$$(X, Y, Z, \quad \dot{Z}) = \text{PTOC}\,[R, \theta, \phi, \dot{R}, \dot{\theta}, \dot{\phi}] \qquad (3.2)$$

Standard superscript arrow notation (A) will be used to designate vectors. Thus, the coordinate transformations become:

$$\vec{P} = \text{CTOP}\,[\vec{C}] \qquad (3.3)$$

and

$$\vec{C} = \text{PTOC}\,[\vec{P}]. \qquad (3.4)$$

On each entry to the event routine, the sequence of events is as follows. The true aircraft position and rate is obtained by invoking the aircraft position routine. These values, which are returned in cartesian coordinates, are transformed to polar coordinates. Any 'adjustments' made for sensor lag and/or regenerative sensor tracking, and/or sensor noise are made to the polar coordinates. The resulting sensed position in polar coordinates is transformed to cartesian coordinates and stored in the SENSED DATA File. The most recent 'n' values stored in the SENSED DATA File are used to compute current values for smoothed position, velocity, and acceleration. These values are also stored in the SENSED DATA File.

The computation for sensor lag is based on true aircraft rates since these are the cause of the phenomenon in the real world. The lag (L) is computed using the equation:

$$\vec{L}_0 = \vec{\dot{P}}_0\left(\frac{1}{K_v} + \frac{1}{K_a \Delta t^2}\right) + \vec{\dot{P}}_{-\Delta t}\left(\frac{-1}{K_a \Delta t^2}\right). \qquad (3.5)$$

This formula is the usual two-term servo lag approximation with acceleration being estimated using two velocity measurements.

The computation of regeneration rates for regenerative sensor tracking is based on data acquired by the sensor. This is again in agreement with the information flow in the real world situation. Estimates of current rates (P_o) are computed by extrapolating smoothed rates computed at the last data acquisition event. Thus:

$$\vec{P}^*_0 = \text{CTOP}\,[\underline{X}_{-\Delta t} + \dot{\underline{X}}_{-\Delta t}\,\Delta t, + \underline{Y}_{-\Delta t} + \dot{\underline{Y}}_{-\Delta t}\,\Delta t, \underline{Z}_{-\Delta t}$$

$$+ \dot{\underline{Z}}_{-\Delta t}\,\Delta t, \dot{\underline{X}}_{-\Delta t}, \dot{\underline{Y}}_{-\Delta t}, \dot{\underline{Z}}_{-\Delta t}] \qquad (3.6)$$

The current estimate is combined with the estimate computed on the previous entry in manner similar to the computation for lag. Thus:

$$\vec{Rg}_0 = \vec{\dot{P}}^*_0\left(\frac{1}{K_v} + \frac{1}{K_a \Delta t^2}\right) + \dot{P}^*_{-\Delta t}\left(\frac{-1}{K_a \Delta t^2}\right). \qquad (3.7)$$

A correlated sequence of random sensor error (noise) is generated as follows: A three-component vector (N_o) is computed using the relationship:

$$\vec{N}_0 = \alpha N_{-\Delta t} + \beta \vec{X} \qquad (3.8)$$

Where: X is a three-component vector of independent normally-distributed variables with mean zero and variance one.

$$k = e^{-\Delta t/.20} \qquad (3.9)$$

$$\alpha = (1 - k^2) \qquad (3.10)$$

$$\beta = (1 - k^2)k \qquad (3.11)$$

This vector is then transformed into random error in range (R), azimuth (θ), and elevation (ϕ) using:

$$N_R = N_1 \sigma_R \qquad (3.12)$$

$$N_\theta = N_2 \sigma_\theta/(R \cos \phi) \qquad (3.13)$$

$$N_\phi = N_3 \sigma_\theta/R. \qquad (3.14)$$

Several computer runs were made using a more complex error equation. This equation is presented in Appendix A.

The full sequence of events in equation form is:

$$\vec{P}'_o = \text{CTOP}\,[\vec{C}_o] + \vec{L}_o + \vec{R}_{go} + \vec{N}_o \qquad (3.15)$$

$$\vec{C}'_o = \text{PTOC}\,[\vec{P}'_o] \quad \text{sensed values} \qquad (3.16)$$

$$\left.\begin{aligned} \underline{\vec{C}}_o &= \sum_{j=0}^{n-1} a_j \vec{C}'_{-j\Delta t} & (3.17)\\ \underline{\dot{\vec{C}}}_o &= \sum_{j=0}^{n-1} b_j \vec{C}'_{-j\Delta t} & (3.18)\\ \underline{\ddot{\vec{C}}}_o &= \sum_{j=0}^{n-1} c_j \vec{C}'_{-j\Delta t} & (3.19) \end{aligned}\right\} \text{smoothed values}$$

On any given run, the inclusions of lag, regeneration, and noise are optional. Further, the number of points to be used in smoothing, as well as the particular coefficients to be used, are also specified at input time.

Some further details are worthy of note. First, since regeneration depends on the presence of smoothed data values for at least two time periods, problems can be encountered if attempts are made to start regenerating rates on the first set of sensed data. Further, since no provision is currently provided for start-up procedures in the data smoother, initial values for smoothed data may be unstable and result in unrealistically large prediction errors. (In a real system the problems would be avoided by providing a start-up procedure in the data smoother.) These problems are avoided by delaying the first regeneration of rates and the first prediction event by one smoothing time.

The preceding comments lead to another important function of this event routine. This event is responsible for the initial scheduling and cancelling of the following events:

a. Prediction.

b. Gun Motion and Firing.

c. Regenerative Tracking Switch.

The Prediction and Gun Motion Events will be scheduled whenever the aircraft is visible to the sensor and they are not currently scheduled. They will be unscheduled whenever the aircraft is not visible and they are currently scheduled. Once they have been scheduled, they will continue to reschedule themselves at their specified period until unscheduled by action of the Sensor Data Acquisition Event.

The Regeneration Switch 'ON' event will be scheduled when the aircraft is first sensed and the Regenerative Switch 'OFF' event will be invoked whenever the aircraft becomes invisible to the sensor (in this case the event is not scheduled but invoked directly from the Sensor Data Acquisition Event).

3.2 REGENERATIVE TRACKING SWITCH EVENT

The function of this event is to turn the regeneration switch 'ON' and 'OFF' (see comments in the description of the Sensor Data Acquisition Event). If regeneration has been specified in the input and it is invoked with an 'ON' signal, it turns the switch 'ON.' If it is invoked with an 'OFF' signal it turns the switch 'OFF.' If regeneration has not been specified in the input, the invocation of this event has no effect on the simulation.

3.3 PREDICTION EVENT

The Prediction Event models that portion of the real gun system in which sensor information is extrapolated into the future in order to compute lead angles for gun pointing. This event routine can operate in two modes. The first, standard mode, makes no restraining assumptions on the aircraft trajectory. The second, defense of a known point, assumes that a known point is the object of attack and biases predictions according to that assumption.

The sequence of computations which occur when an entry is made to this event routine are as follows: First, the Time of Flight Module is invoked which, with the aid of the Prediction Algorithm Module, computes a time of flight. The returned value for time of flight is combined with current time and the Prediction Algorithm Module is invoked directly to obtain the required predicted position. At this point, if standard prediction has been specified, a branch is executed to the area which stores data in the PREDICTED POSITION File. If defense of a known point has been specified, the process continues.

As mentioned above, when defense of a known point is specified, predictions are biased. The bias may take two forms based on either the assumption that the aircraft will turn toward the defended point, or the assumption that the aircraft will follow a straight line

from its present position to the defended point. Consideration of the geometry of the situation reveals that neither of these biases are appropriate under certain conditions. For this reason, even when bias is specified, a particular prediction may or may not be biased. The determining factor is the position and velocity of the aircraft relative to the defended point. If the defended point is in front of the aircraft, a bias is considered. 'In front of' should be understood as an angle of less than 90 degrees between the aircraft horizontal velocity vector and the horizontal projection of the vector from the aircraft to the defended point. Since the dot product of two vectors is positive, if and only if the absolute value of the angle between the vector involved is less than 90 degrees, the decision rule becomes:

a. Consider bias if: $V.R > 0$.

b. Do not consider bias if: $F.R \leq 0$.

Figure 3-1 displays the geometry of the decision rule. If the defended point lies within the shaded area bias is considered.

Whenever the defended point lies within the bias consideration area of the aircraft, the geometry is further examined to determine if a bias can be used, and if so, which of the two biases is appropriate. Three situations are possible; (1) the aircraft heading is such that a projection of its current velocity vector would pass within a specified (during initialization) distance (R_o) of the defended point, (2) a standard turn of specified (during initialization) acceleration followed by a straight line flight path would cause the aircraft to pass over the target, or (3) neither of the above.

The first situation, namely the straight line bias, is the easiest to explain (see Figure 3-2). The heading angle change θ is computed as:

$$\theta = \mathrm{Tan}^{-1}\left(\frac{R \times V}{R \cdot V} - 1/2\,\dot{\theta}t\right) \qquad (3.20)$$

The closest approach to the defended point is approximated by the arc:

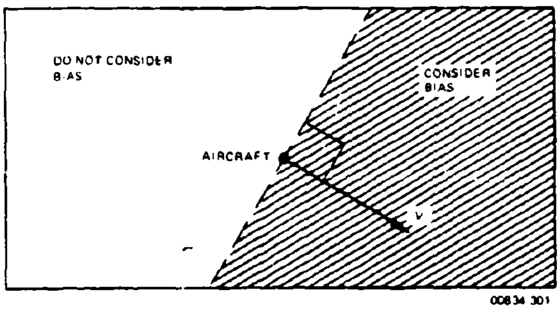


Figure 3-1. Bias Consideration Region

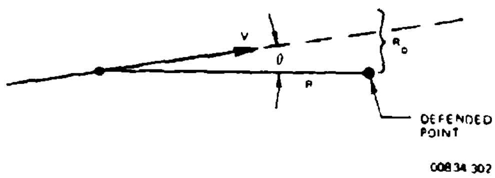


Figure 3-2. Straight-Line Bias Situation

$$S = R\theta \qquad (3.21)$$

Linear bias is used if $S \leq R_o$.

The linear-bias computations are accomplished in the Time of Flight Module and the Prediction Algorithm Module. If a situation is such that linear bias is appropriate, an indicator switch is set and point to point interpolation will be used in the Prediction Algorithm Module.

If the situation is such that linear bias is not appropriate, a check is made to see if a turn followed by a straight-line path can bring the aircraft over the target. This check is made as follows: The radius of turn is computed using:

$$\delta = V^2/ng \qquad (3.22)$$

where:

V = the aircraft velocity

ng = the standard turn acceleration (n is input)

The heading and turn radius are used to find the center of curvature of the turn and the distance (L) from this point to the defended point. Whenever the turn radius is less than the distance from the center of curvature, a turn is possible (Figure 3-3). If no turn is possible, we have situation three: that is, no bias can be used (Figure 3-4). Under such circumstances a branch to the data storage area is executed and standard values for prediction are used.

If a turn followed by a linear segment is to be used, the process continues. Figure 3-5 presents the detailed geometry of the turn. The new heading (θ) and the time to turn (T_T) are computed using the following formulas:

$$R_p = \sqrt{R_o^2 - 2\delta\, R_o \sin H} \qquad (3.23)$$

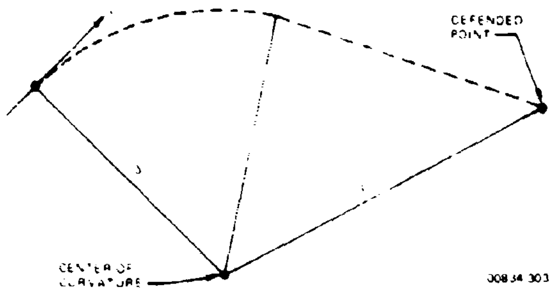


Figure 3-3. Turn to Attack Bias Situation (S < L)

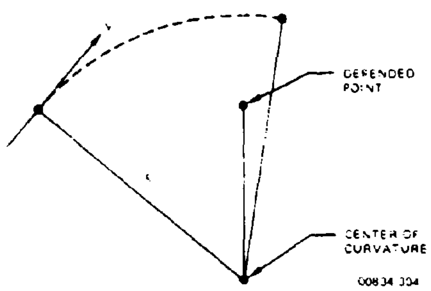


Figure 3-4. No Bias Situation (S ≥ L)

$$A = \mathrm{Tan}^{-1} \left\{ \frac{(R_o - \delta \sin H) - R_p \delta \cos H}{R_p (R_o - \delta \sin H) + \delta^2 \cos H} \right\} \quad (3.24)$$

$$\Delta\theta = A + H \quad (3.25)$$

$$\dot{\theta} = \frac{V}{S} \quad (3.26)$$

$$T_T = \theta / \dot{\theta} \quad (3.27)$$

$$\theta = \theta_o + \Delta\theta \quad (3.28)$$

cluded. The time of flight to the end point of the turn is compared to the time to turn. If the time of flight exceeds the time to turn, the biased prediction is used, otherwise a branch to the storage area is made and a standard prediction results.

When turn bias is included, values for the velocity in the x and y direction are computed. The prediction in these directions is a linear extrapolation along the line from the end of the turn to the defended point. In this case, the Z coordinate is unbiased.

For all cases, the last action is a branch to the data

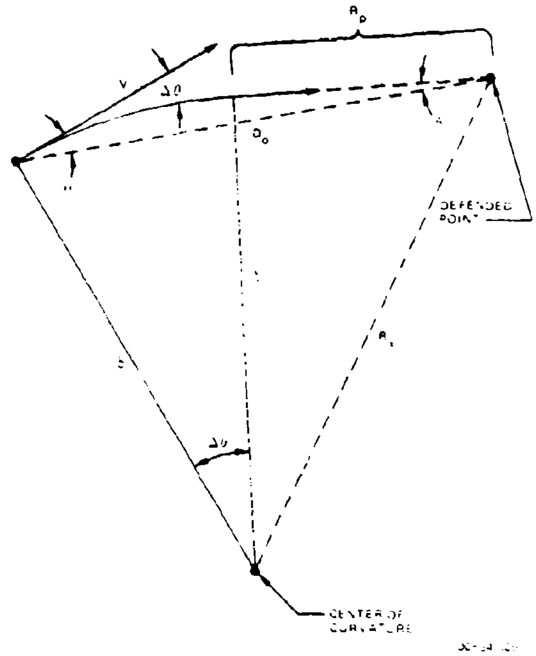


Figure 3-5. Geometry of a Turn Followed by a Linear Segment

storage area where the predicted position is transformed to polar coordinates and stored along with the estimated time of closest approach as the most recent entry in the PREDICTED POSITION File.

Figure 3-6 summarizes the logical structure of the Prediction Event.

3.4 GUN MOTION AND FIRING EVENT

The function of this event is to compute the closest approach of a shell fired at current clock time given: the history of predicted positions, the ballistic coefficients, and the actual aircraft flight path. Gun servo lag and/or servo regeneration can be included. When included, lag and/or regeneration characteristics are determined by input parameter specification.

In the following discussion, upper case letters are used to designate predicted values, subscripts designate the time to which the values pertain, primed upper case letters are used to indicate actual gun pointing angles, and the letters L and R are used to designate Lag and Regeneration respectively. Standard dot notation is used to indicate rates and accelerations. In this notation, the azimuth and elevation become:

$$\theta'_o = \theta_o + L_o + R_o \quad (3.29)$$

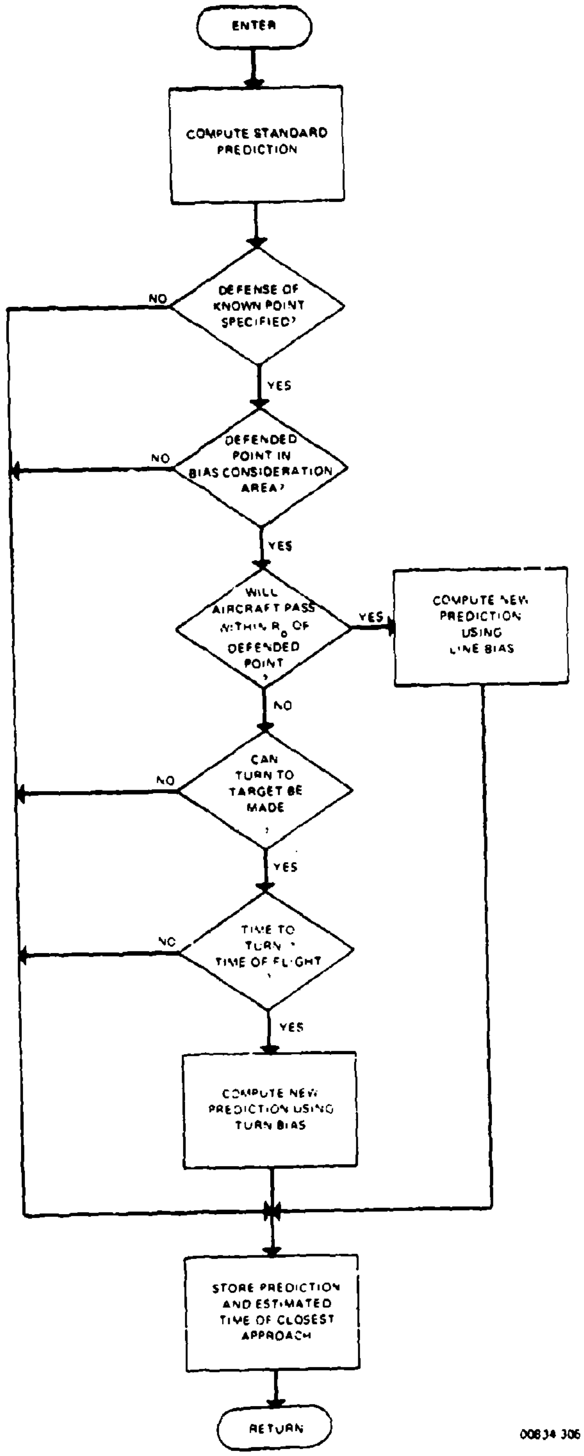


Figure 3-6. Logic Flow Diagram of the Prediction Event

$$\phi_0' = \phi_0 + L_0 + R_0 \tag{3.30}$$

The lag computation is based on the four latest predicted positions. These four values are used to estimate current rate and acceleration values for the coordinate in question, which is in turn used in the expression:

$$L_A = \frac{\dot{A}}{K_v} + \frac{\ddot{A}}{K_g} \tag{3.31}$$

When the estimates are substituted into the above expression the lag equations become:

$$L_A = \left\{\frac{1.0}{K_v \Delta t} + \frac{2.0}{K_a \Delta t^2}\right\} A_0 + \left\{\frac{-1.0}{K_v \Delta t} + \frac{-5.0}{K_a \Delta t^2}\right\} A_{-\Delta t} \tag{3.32}$$

$$+ \left\{\frac{4.0}{K_a \Delta t^2}\right\} A_{-2\Delta t} + \left\{\frac{-1.0}{K \Delta t^2}\right\} A_{-3\Delta t} \tag{3.33}$$

Values for regeneration are computed as follows: The current predicted position is transformed to cartesian coordinates. These are combined with current values for smoothed cartesian rates and transformed back to polar coordinates. The resulting polar rates are used to estimate polar acceleration using the following equations:

$$S = \sqrt{X_p^2 + Y_p^2} \tag{3.34}$$

$$\dot{S} = (X_p \dot{X}_p + Y_p \dot{Y}_p)/S \tag{3.35}$$

$$\ddot{\theta} = -2.0\, \dot{S}\dot{\theta}/D \tag{3.36}$$

$$\ddot{\phi} = \frac{-2.0\, D\, \dot{D}\dot{\phi} - S\dot{\theta} Z_p}{D^2} \tag{3.37}$$

These values are used in the standard two-term lag estimate yielding:

$$R_\theta = \left\{\frac{\dot{\theta}}{K_v \Delta t} + \frac{\ddot{\theta}}{K_a \Delta t^2}\right\} \tag{3.38}$$

$$R_\phi = \left\{\frac{\dot{\phi}}{K_v} + \frac{\ddot{\phi}}{K_a}\right\} \tag{3.39}$$

The projectile position, at current time plus time of flight (estimated time of closest approach), is given by: the predicted range (D_o), the adjusted azimuth (θ'_c), and the adjusted elevation (ϕ_c). The projectile velocity

is assumed to be in the radial direction with magnitude given by:

$$V_p(t) = \left|\frac{a-1}{K} V_m^{a-1} t + 1.0\right|^{\frac{-1}{a-1}} V_m \quad (3.40)$$

where:

t = time of flight.

V_m = muzzle velocity.

a, K = ballistic coefficients from the Time of Flight module.

Projectile position and velocity are transformed into cartesian coordinates for the computation of the actual closest approach values.

The time of closest approach is computed by correcting the estimated time of closest approach which is computed by the predictor. The correction is made by assuming: that the aircraft and projectile are moving at constant rates for the time period necessary for the projectile to get from either the predicted position to position of closest approach, or in the case when the closest approach occurs earlier than the predicted time, the time necessary to travel from the closest approach to the predicted position. Once this assumption is made, the time from predicted position to closest approach can be computed as:

$$t_c = |\overline{R} \times \overline{V}| / \overline{V} \cdot \overline{V} \quad (3.41)$$

where:

$\overline{R}$ = position of aircraft relative to the projectile at predicted impact.

$\overline{V}$ = velocity of the aircraft relative to the projectile at the predicted impact.

The time of closest approach is then given by:

$$t = t_p + t_c \quad (3.42)$$

The position of the shell at closest approach is given by:

$$\overline{R}_s = \overline{R}_p + \overline{V}_s t_c \quad (3.43)$$

If the error components are desired, they are computed as:

$$\overline{E} = \overline{R} + \overline{V}\, t_c \quad (3.44)$$

With $\overline{R}$, $\overline{V}$, and t_c defined as above.

The time of fire, the range, and the miss distance at closest approach are stored in the SHELL POSITION File. There are several versions of this event routine which differ in the contents stored in the file for printing.

SECTION 4
UTILITY PACKAGE

4.1 AIRCRAFT POSITION ROUTINE

The Aircraft Position Routine provides the position and velocity of the aircraft in cartesian coordinates centered at the gun for the simulation time specified in the invoking command.

Paths are represented as sequences of continuous path segments proceeding from an initial point of known position and velocity. Each path segment is either a segment of a line, a segment of a circle, or a segment of a logarithmic spiral. All path segments are in the horizontal plane coupled with motion of constant acceleration (possibly zero) in the vertical direction. Smooth positions and velocities are assured by using the end conditions of one segment as the initial conditions of its successor. The end point of a segment is determined relative to its initial point with the 'duration' of the segment being specified by: length for a line segment, and degrees of turn for both circular and spiral segments. The only position and velocity specification made is for the initial point. Further path description is in the form of the following statements:

a. Line 'X' meters in length, tangential acceleration 'A_T,' and vertical acceleration 'A_Z.'

b. Circular turn 'ϕ' degrees right or left radial acceleration 'A_R' and vertical acceleration 'A_Z.'

c. Spiral turn 'ϕ' degrees right or left, radial acceleration 'A_R,' tangential acceleration 'A_T,' and vertical acceleration 'A_Z.'

For linear segments position and velocity are determined according to the equations:

$$X(t) = X_o \cdot \left(V_o t + \tfrac{1}{2} A_T t^2\right) \cos\theta \tag{4.1}$$

$$Y(t) = Y_o + \left(V_o t + \tfrac{1}{2} A_T t^2\right) \sin\theta \tag{4.2}$$

$$Z(t) = Z_o + V_{Z_o} t + \tfrac{1}{2} A_Z t^2 \tag{4.3}$$

$$V_X(t) = -(V_o + A_T t) \cos\theta \tag{4.4}$$

$$V_Y(t) = (V_o + A_T t) \sin\theta \tag{4.5}$$

$$V_Z(t) = V_{Z_o} + A_Z t \tag{4.6}$$

Where:

θ = aircraft heading

For circular segments, position and velocity are determined according to the equations:

$$R = V_o^2 / A_R \tag{4.7}$$

$$\dot{\theta} = V_o / R \tag{4.8}$$

$$\theta(t) = \theta_o + \dot{\theta} t \tag{4.9}$$

$$X(t) = X_o + R\,[\cos\theta_o - \cos\theta(t)] \tag{4.10}$$

$$Y(t) = Y_o + R\,[\sin\theta_o + \sin\theta(t)] \tag{4.11}$$

$$Z(t) = Z_o + V_{Z_o} t + \tfrac{1}{2} A_Z t^2 \tag{4.12}$$

$$V_X(t) = -V \sin\theta(t) \tag{4.13}$$

$$V_Y(t) = V \cos\theta(t) \tag{4.14}$$

$$V_Z(t) = V_{Z_o} + A_Z t \tag{4.15}$$

For spiral segments position and velocity are determined according to the equations:

$$\lambda = A_T / A_R \tag{4.16}$$

$$V(t) = V_o + A_T t \tag{4.17}$$

$$R = V_o^2 / A_R \tag{4.18}$$

$$\theta(t) = \theta_o - R \log_e [V^2(t) / 2\lambda V_o^2] \tag{4.19}$$

$$X(t) = X_o - \frac{V^2(t)}{A_R(1 + 4\lambda^2)} \left\{ 2\lambda \left[\sin\theta_o - \frac{V^2(t)}{V_o^2} \sin(\theta(t)) \right] - \left[\cos\theta_o - \frac{V^2(t)}{V_o^2} \cos(\theta(t)) \right] \right\} \tag{4.20}$$

$$Y(t) = Y_o = \frac{V^2(t)}{A_R(1 + 4\lambda^2)} \left\{ 2\lambda \left[\frac{V^2(t)}{V_o^2} \cos\theta(t) - \cos\theta_o \right] - \left[\sin\theta_o - \frac{V^2(t)}{V_o^2} \sin\theta_o \right] \right\} \tag{4.21}$$

$$Z(t) = Z_0 + V_{Z_0}t + \frac{1}{2}A_Z t^2 \quad (4.22)$$

$$V_X(t) = -V(t)\sin[\theta(t)] \quad (4.23)$$

$$V_Y(t) = V(t)\cos[\theta(t)] \quad (4.24)$$

$$V_Z(t) = V_{Z_0} + A_Z t \quad (4.25)$$

4.2 TIME OF FLIGHT MODULE

The purpose of the Time of Flight Module is to compute the time required for a projectile to travel from the gun to a future aircraft position. The problem is somewhat complicated by the fact that the predicted position is dependent on the time which is of course based on the time of flight. Closed-form solutions of the problem are possible only under some rather restrictive assumptions. For this reason numerical methods have been employed. The time of flight computation is based on the fact; that at the time of predicted impact, the range to the aircraft (R_a) and to the projectile (R_p) must be equal. In essence then, the Time of Flight Module numerically solves the equation:

$$f(t) = R_a(t) - R_p(t) = 0 \quad (4.26)$$

for the smallest value of t.

The values of aircraft range are provided by the Prediction Algorithm Module and are dependent on the values currently stored in the SENSED DATA File as well as the particular options for prediction being simulated. Values for $R_p(t)$ are computed using the equation:

$$R_p(t) = a[(bt + 1)^c - 1] \quad (4.27)$$

where:

$a = 1.0\ K(\alpha - 2)\ V_m^{\alpha-2}$

$b = K(\alpha - 1)\ V_m^{\alpha-1}$

$c = (\alpha - 2)/(\alpha - 1)$

α = input variable

K = input variable

V_m = input variable

A start value for the solution is obtained by assuming a projectile which does not decelerate, and an aircraft which is in unaccelerated motion:

$$T_{start} = R_a(0)\left[\dot{R}_a(0) + \sqrt{\dot{R}_a^2(0) + V_m^2 - V_a^2(0)}\right] / \left(V_m^2 - V_a^2(0)\right) \quad (4.28)$$

where:

V_m = muzzle velocity of gun.

V_a = current aircraft velocity.

$\dot{R}_a$ = current range rate of the aircraft.

An initial computation of the function $f(t) = R_a(t) - R_p(t)$ is made using the start value. The process is continued with each successive computation being made at a new time which is the current value incremented by a function of the start value.

Once a sign change has been obtained, a combination of the secant method and interval bisection is employed until the desired accuracy is obtained. In general, no more than five computations are required to derive the required accuracy.

4.3 PREDICTION ALGORITHM MODULE

The function of the Prediction Algorithm Module is to extrapolate current available aircraft position and velocity data to a specified future time point; thereby, producing an estimate of aircraft position for the specified time. The module has the capability of performing the extrapolation by a number of methods; each of which is somewhat flexible in that its specific behavior is determined by method parameters. The specific method, employed during a simulation run, is determined by the designation of the general method type and its required parameters during the initialization phase of the run. The position data on which the module operates are all derived from the output of the sensor module. These data are accessed via the communications pool (specifically the SENSED DATA File).

Current module capabilities include three general methods:

a. Quadratic (includes linear by parameter specification).

b. Simple Polar.

c. Defense of a Known Point (in combination with Quadratic).

4.3.1 Quadratic Prediction

The quadratic prediction is made using smoothed position (X, Y, Z), velocity (X, Y, Z), and acceleration (X, Y, Z) values computed and stored in the SENSED

POSITION File by the TRACK Module. The prediction is computed as: a linear estimate corrected for accelerations in the horizontal plane, and changes in total aircraft speed induced by changes in aircraft altitude. The basic estimate for a time increment τ is given by:

$$X' = \underline{X} + \dot{\underline{X}}\tau \qquad (4.29)$$

$$Y' = \underline{Y} + \dot{\underline{Y}}\tau \qquad (4.30)$$

$$Z' = \underline{Z} + \dot{\underline{Z}}\tau \qquad (4.31)$$

The correction for acceleration in the horizontal plane is made whenever the absolute value of the estimated heading rate θ of the aircraft exceeds the value of an input threshold parameter. The specification of an excessively large threshold value leads to suppression of the acceleration correction and linear prediction results. The heading rate is estimated by the equation:

$$\dot{\underline{\theta}} = (\ddot{\underline{X}}\dot{\underline{Y}} - \dot{\underline{X}}\ddot{\underline{Y}})/(\dot{\underline{X}}^2 + \dot{\underline{Y}}^2) \qquad (4.32)$$

When the correction is included, the resultant predictions for X and Y are:

$$X' = \underline{X} + \dot{\underline{X}}\tau - \frac{1}{2}\dot{\underline{\theta}}\left(\tau^2 + \tau\Delta t + \frac{\Delta t^2}{6}\right)\dot{\underline{Y}} \qquad (4.33)$$

$$Y' = \underline{Y} + \dot{\underline{X}}\tau + \frac{1}{2}\dot{\underline{\theta}}\left(\tau^2 + \tau\Delta t + \frac{\Delta t^2}{6}\right)\dot{\underline{X}} \qquad (4.34)$$

The correction for change in aircraft speed, due to a change in altitude, is made whenever the absolute value of the aircraft velocity in the vertical direction exceeds the value of an input threshold parameter. As in the case of the acceleration, the correction can be suppressed by specification of an excessive value for the parameter. When the correction is included, the resultant predictions are:

$$X' = \underline{X} + \dot{\underline{X}}\tau - \frac{1}{2}\left(\tau^2 + \tau\Delta t + \frac{\Delta t^2}{6}\right) \quad [\dot{\underline{\theta}}\dot{\underline{Y}} + g(\dot{\underline{X}}\dot{\underline{Z}}/V^2)] \qquad (4.35)$$

$$Y' = \underline{Y} + \dot{\underline{Y}}\tau + \frac{1}{2}\left(\tau^2 + \tau\Delta t + \frac{\Delta t^2}{6}\right) \quad [\dot{\underline{\theta}}\dot{\underline{X}} - g(\dot{\underline{Y}}\dot{\underline{Z}}/V^2)] \qquad (4.36)$$

$$Z' = \underline{Z} + \dot{\underline{Z}}\tau + \frac{1}{2}\left(\tau^2 + \tau\Delta t + \frac{\Delta t^2}{6}\right) \quad [g(\dot{\underline{Z}}^2/V^2)] \qquad (4.37)$$

where:

Δt = Sample Interval

V = Estimated Aircraft Velocity

g = Acceleration due to gravity

4.3.2 Simple Polar Prediction

The Simple Polar Prediction is made by estimating the range rate R, the azimuth rate A, the elevation rate E; assuming constant rates, and using linear extrapolation. Polar coordinate rates are estimated by transforming the cartesian coordinates of the last two entries in the SENSED POSITION File and computing first-divided differences. Thus:

$$\dot{\underline{R}} = (\underline{R}_t - \underline{R}_{t-\Delta t})/\Delta t \qquad (4.38)$$

$$\dot{\underline{A}} = (\underline{A}_t - \underline{A}_{t-\Delta t})/\Delta t \qquad (4.39)$$

$$\dot{\underline{E}} = (\underline{E}_t - \underline{E}_{t-\Delta t})/\Delta t \qquad (4.40)$$

Estimated range, azimuth, and elevation are given by:

$$R' = \underline{R}_t + \dot{\underline{R}}\tau \qquad (4.41)$$

$$A' = \underline{A}_t + \dot{\underline{A}}\tau \qquad (4.42)$$

$$E' = \underline{E}_t + \dot{\underline{E}}\tau \qquad (4.43)$$

The required cartesian estimates are obtained by transforming R', A', and E'.

4.3.3 Defense of a Known Point

The majority of the logic and computation for the defense of a known point option are in the Prediction Event Module. However, for time of flight computations, use is made of the Prediction Algorithm Module. During such entries to the module, one of two modes of prediction are possible: Quadratic (discussed previously) or point-to-point linear interpolation. An indicator switch set in the Prediction Event Module controls which method is used on any given entry to the module. The following pertains to point-to-point linear interpolation used for a portion of the attack of a known point option.

Figure 4-1 presents the geometry for the point-to-point linear interpolation. The point Q represents the last sensed aircraft position and P the point being defended. The prediction method assumes that the aircraft will travel along the line connecting the points at rates which are consistent in magnitude to the current smoothed aircraft speed. Distance along the line is computed using the equations:

$$\Delta X = X_P - X_Q \qquad (4.44)$$

$$\Delta Y = Y_P - Y_Q \qquad (4.45)$$

$$\Delta Z = Z_P - Z_Q \qquad (4.46)$$

$$D = \sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2} \qquad (4.47)$$

$$L(t) = Vt + \frac{1}{2}g\left(t^2 + tT_s + \frac{T_s^2}{6}\right)\frac{\Delta Z}{D} \qquad (4.48)$$

The desired estimate is then obtained by interpolation:

$$X = X_Q + \Delta X \cdot L(t)/D \qquad (4.49)$$

$$Y = Y_Q + \Delta Y \cdot L(t)/D \qquad (4.50)$$

$$Z = Z_Q + \Delta Z \cdot L(t)/D \qquad (4.51)$$

4.4 RANDOM NUMBER GENERATOR

The function of the Random Number Generator is to produce samples of random variables from either a population uniformly distributed on the interval [0,1], or from a population which is normally distributed with mean zero and variance one. The uniform random variables are generated by using an additive congruous process with 31 seeds. This method was based on work which was done at MIT's Lincoln Laboratory in 1958 by B.F. Green, J.E. Smith, and L. Klem. The sequence of samples is produced by using the following equations:

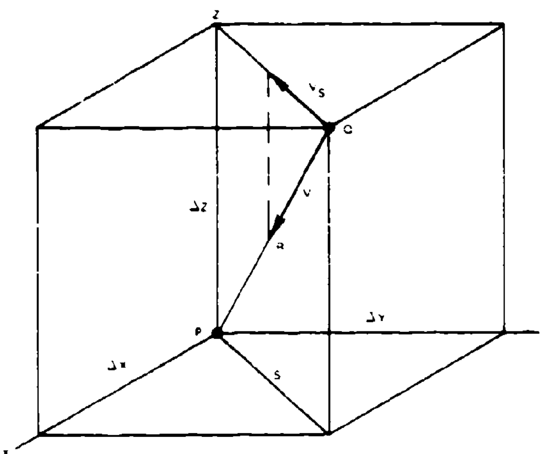


Figure 4-1 Geometric Representation of the Point-to-Point Linear Interpolation

$$X_j = (X_{j-1} + X_{j-n}) \text{ Mod } 2^r \qquad j \geq n \qquad (4.52)$$

$$R_{j-n+1} = X_j/2^r \qquad (4.53)$$

where:

$n = 31$

$r = 29$

X_1 = an input seed variable

$X_2 - X_{31}$ = prestored seed values

R_j = the required uniformly distributed random numbers.

Normally-distributed variables are computed from the uniformly-distributed random variables using the well known Central Limit Theorem. The exact equation employed is:

$$N = \sum_{i=1}^{12} R_i - 6.0 \qquad (4.54)$$

where:

N = the required sample

R_i = uniformly distributed random variables

This module has the ability to return any number of normal or uniform numbers on any given entry. The sequence of samples generated is a function of the first seed value which is an input variable.

4.5 COORDINATE TRANSFORMATION

There are two coordinate transformation programs. One converts polar coordinates to cartesian coordinates and the other converts cartesian coordinates to polar coordinates. Both programs convert the three position coordinates plus their respective rates. The transformations are not standard due to the fact that the azimuth angle is measured counter-clockwise from the positive y-axis. This system was used in an attempt to remain as compatible as possible with systems employed in the University of Michigan Gun Model.

The cartesian to polar transformation uses Equations 4.55 through 4.61. The equations are computed in the order shown and earlier results are used in later computation in order to minimize computer time.

$$R = \sqrt{X^2 + Y^2 + Z^2} \qquad (4.55)$$

$$S = \sqrt{X^2 + Y^2} \qquad (4.56)$$

$$\theta = \mathrm{Tan}^{-1}(-X/Y) \quad (4.57)$$

$$\phi = \mathrm{Tan}^{-1}(Z/S) \quad (4.58)$$

$$\dot{R} = (X\dot{X} + Y\dot{Y} + Z\dot{Z})/R \quad (4.59)$$

$$\dot{\theta} = (X\dot{Y} - \dot{X}Y)/S^2 \quad (4.60)$$

$$\dot{\phi} = (Z - Z\dot{R}/R)/S \quad (4.61)$$

The polar to cartesian conversion uses the following equations:

$$S = R\cos\phi \quad (4.62)$$
$$Z = R\sin\phi \quad (4.63)$$
$$X = -S\sin\theta \quad (4.64)$$
$$Y = S\cos\theta \quad (4.65)$$
$$T = \dot{R}/R - Z\dot{\phi}/S \quad (4.66)$$
$$\dot{X} = XT - Y\dot{\theta} \quad (4.67)$$
$$\dot{Y} = X\dot{\theta} + YT \quad (4.68)$$
$$\dot{Z} = Z\dot{R}/R + S\dot{\phi} \quad (4.69)$$

4.6 REPORT GENERATORS

Report generators are provided to compute and output measures of effectiveness for the gun system being simulated. Due to the variability of the measures required for different aspects of the system study, two general types of report generators have been provided. For detailed analyses of the effect of a single aspect of the system, periodic reports are provided. For the comparison of the effectiveness of the system as a whole, summary reports are produced by an exit report generator.

4.6.1 Periodic Reports

Periodic Reports are generated by special report events. They may be either scheduled or invoked by another module. In general, they have access to any data in the communication pool but may not alter these values. Figure 4-2 presents an example of the output from a periodic report.

This report presents a description of the aircraft and shell positions at the time of closest approach of the shell to the aircraft. The time entries include: the clock time when the shell was fired, the estimated time of flight, and the clock time at the closest approach.

The first line presents shell range in meters, azimuth in mils, and elevation in mils. The second line presents aircraft range in meters, azimuth in mils, and elevation in mils. The third line presents the difference in position (measured in meters) in the range, azimuth and elevation directions. The fourth line presents the errors in the azimuth and elevation coordinates measured in mils. This particular report was generated at a time interval of 0.20 second.

4.6.2 Exit Report Generator

As stated previously, the exit report generator is concerned with the performance of the system as a whole. The measures of effectiveness which it produces are therefore biased towards integrations of performance over the entire duration of the course. Other measures may be produced and outputted by indicating the requirement to do so on a print options card.

With no output indicated, the exit report will generate for each replication:

$$K_s = \int_0^T \mathrm{pss}\,dt \quad (4.70)$$

where:

pss = single shot kill probability

T = duration of the aircraft flight

In addition, at the end of the last replication the mean and standard deviation of K_s over the replication is output. Furthermore, the distribution over the replicates of the average pss over bursts, and the distribution of miss distances for single samples are output. Figure 4-3 presents an example of the summary produced by the exit report.

Information may also be output at the individual replicate, burst, or sample level. Figures 4-4 through 4-7 present examples of the outputs at these various levels.

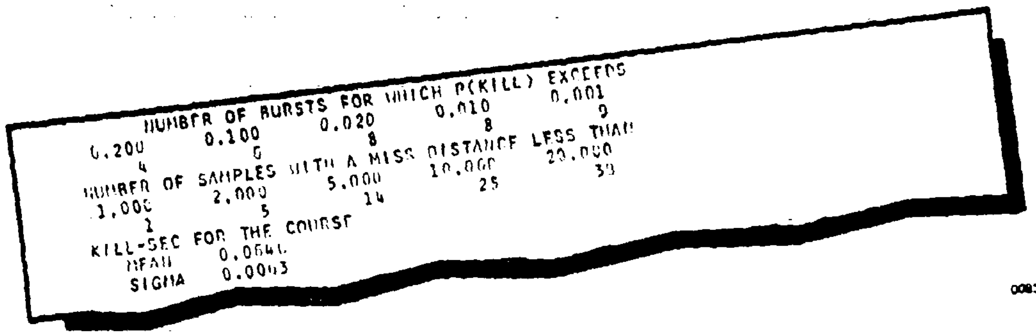
Additional Outputs

In any particular study for which the current outputs are not suitable, additional reports can be generated with minor modification to the model. In most cases, the modifications will be the replacement of an existing report subroutine with one more appropriate to the study being conducted.

CLOCK	TIME FLIGHT	PREDICTED	RANGE	POSITION AZIMUTH	ELEVATION
8.40	0.57	8.97	618.03	-473.94	425.05
			616.05	-480.22	425.76
		ERROR(METERS)	1.97	3.48	-0.43
		ERROR(MILS)		6.27	-0.72
8.60	0.55	9.15	589.74	-411.42	443.50
			596.44	-416.47	440.69
		ERROR(METERS)	-6.70	2.65	1.63
		ERROR(MILS)		5.05	2.81
8.80	0.53	9.33	562.72	-345.77	462.20
			579.48	-346.09	454.50
		ERROR(METERS)	-16.76	0.16	4.29
		ERROR(MILS)		0.32	7.76
9.00	0.51	9.51	541.92	-274.84	478.81
			565.59	-269.47	466.50
		ERROR(METERS)	-23.66	-2.54	6.55
		ERROR(MILS)		-5.37	12.31
9.20	0.50	9.70	526.60	-192.09	490.59
			555.02	-186.06	476.09
		ERROR(METERS)	-28.42	-2.76	7.49
		ERROR(MILS)		-6.03	14.55
9.40	0.50	9.90	521.81	-101.36	495.43
			548.27	-97.20	482.43
		ERROR(METERS)	-26.45	-1.86	6.66
		ERROR(MILS)		-4.17	13.00
9.60	0.50	10.10	532.25	-7.56	492.31
			545.72	-4.26	484.86
		ERROR(METERS)	-13.47	-1.52	3.89
		ERROR(MILS)		-3.29	7.45
9.80	0.50	10.30	542.96	85.79	486.35
			547.46	90.97	483.29
		ERROR(METERS)	-4.56	-2.45	1.95
		ERROR(MILS)		-5.18	3.65
10.00	0.50	10.50	553.91	175.89	478.79
			553.24	185.99	477.74
		ERROR(METERS)	0.67	-4.89	0.57
		ERROR(MILS)		-10.16	1.05
10.20	0.51	10.71	568.36	264.64	466.79
			563.16	279.37	468.67
		ERROR(METERS)	5.2[illegible]	-7.37	-1.05
		ERROR(MILS)		-14.73	-1.86
10.40	0.53	10.93	579.62	353.26	454.62
			577.25	369.63	456.38
		ERROR(METERS)	1.77	-8.38	-1.9[illegible]
		ERROR(MILS)		-16.35	-1.76

Figure 4-2 Typical Periodic Report

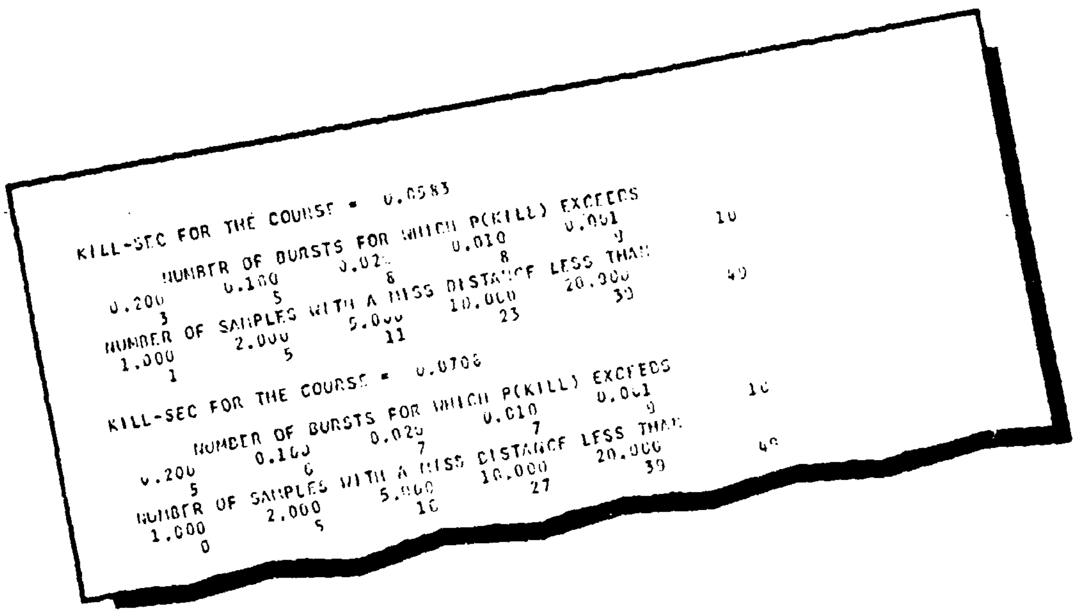
00834-402



```
NUMBER OF BURSTS FOR WHICH P(KILL) EXCEEDS
0.200     0.100     0.020     0.010     0.001
  4         6         8         8         9
NUMBER OF SAMPLES WITH A MISS DISTANCE LESS THAN
1.000     2.000     5.000    10.000    20.000
  1         5        14        25        39
KILL-SEC FOR THE COURSE
  MEAN    0.0641
  SIGMA   0.0043
```

00834-403

Figure 4-3. Typical Summary Exit Report



```
KILL-SEC FOR THE COURSE =  0.0583
  NUMBER OF BURSTS FOR WHICH P(KILL) EXCEEDS
0.200     0.100     0.020     0.010     0.001
  3         5         6         8         9         10
NUMBER OF SAMPLES WITH A MISS DISTANCE LESS THAN
1.000     2.000     5.000    10.000    20.000
  1         5        11        23        39         40
KILL-SEC FOR THE COURSE =  0.0706
  NUMBER OF BURSTS FOR WHICH P(KILL) EXCEEDS
0.200     0.100     0.020     0.010     0.001
  5         6         7         7         9         10
NUMBER OF SAMPLES WITH A MISS DISTANCE LESS THAN
1.000     2.000     5.000    10.000    20.000
  0         5        10        27        39         40
```

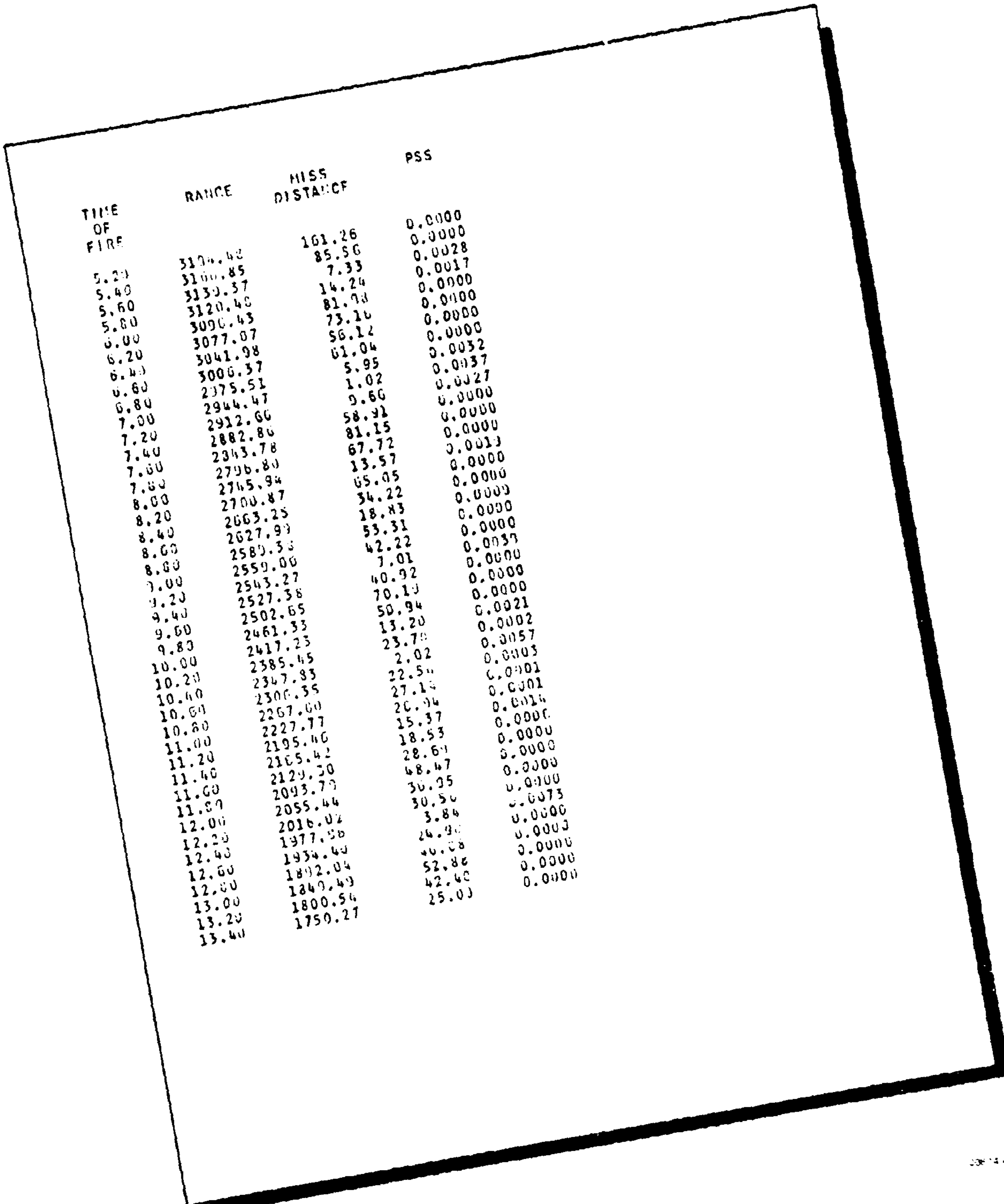
00834-404

Figure 4-4. Typical Replicate-Level Output Report

E(KILLING HITS) = 0.04	P(KILL) =0.042
E(KILLING HITS) = 0.07	P(KILL) =0.064
E(KILLING HITS) = 0.04	P(KILL) =0.043
E(KILLING HITS) = 0.01	P(KILL) =0.009
E(KILLING HITS) = 0.06	P(KILL) =0.056
E(KILLING HITS) = 0.06	P(KILL) =0.059
E(KILLING HITS) = 0.02	P(KILL) =0.019
E(KILLING HITS) = 0.07	P(KILL) =0.068
E(KILLING HITS) = 0.10	P(KILL) =0.092
E(KILLING HITS) = 0.13	P(KILL) =0.118
E(KILLING HITS) = 0.14	P(KILL) =0.127
E(KILLING HITS) = 0.37	P(KILL) =0.310
E(KILLING HITS) = 0.37	P(KILL) =0.313
E(KILLING HITS) = 1.04	P(KILL) =0.648
E(KILLING HITS) = 3.94	P(KILL) =0.980
E(KILLING HITS) = 1.09	P(KILL) =0.665
E(KILLING HITS) = 4.99	P(KILL) =0.993
E(KILLING HITS) = 0.91	P(KILL) =0.597
E(KILLING HITS) = 0.39	P(KILL) =0.324
E(KILLING HITS) = 0.18	P(KILL) =0.164
E(KILLING HITS) = 0.37	P(KILL) =0.309
E(KILLING HITS) = 0.24	P(KILL) =0.210
E(KILLING HITS) = 0.13	P(KILL) =0.121
E(KILLING HITS) = 0.04	P(KILL) =0.044

KILL-SEC FOR THE COURSE = 0.3662

Figure 4-5 Typical Burst-Level Output Report



TIME OF FIRE	RANGE	MISS DISTANCE	PSS
5.20	3194.42	161.26	0.0000
5.40	3166.85	85.56	0.0000
5.60	3139.37	7.33	0.0028
5.80	3120.40	14.24	0.0017
6.00	3090.43	81.98	0.0000
6.20	3077.07	73.16	0.0000
6.40	3041.98	56.12	0.0000
6.60	3006.37	61.04	0.0000
6.80	2975.51	5.95	0.0032
7.00	2944.47	1.02	0.0037
7.20	2912.66	9.66	0.0027
7.40	2882.86	58.91	0.0000
7.60	2843.78	81.15	0.0000
7.80	2796.80	67.72	0.0000
8.00	2745.94	13.57	0.0013
8.20	2700.87	65.05	0.0000
8.40	2663.25	34.22	0.0000
8.60	2627.99	18.83	0.0009
8.80	2589.33	53.31	0.0000
9.00	2559.00	42.22	0.0000
9.20	2543.27	7.01	0.0037
9.40	2527.38	40.92	0.0000
9.60	2502.65	70.19	0.0000
9.80	2461.33	50.94	0.0000
10.00	2417.23	13.20	0.0021
10.20	2385.45	23.79	0.0002
10.40	2347.83	2.02	0.0057
10.60	2306.35	22.54	0.0003
10.80	2267.60	27.14	0.0001
11.00	2227.77	26.94	0.0001
11.20	2195.46	15.37	0.0014
11.40	2165.42	18.53	0.0000
11.60	2129.30	28.69	0.0000
11.80	2093.79	48.47	0.0000
12.00	2055.44	30.35	0.0000
12.20	2016.02	30.56	0.0000
12.40	1977.56	3.84	0.0073
12.60	1934.40	24.96	0.0000
12.80	1892.04	40.68	0.0003
13.00	1849.49	52.86	0.0000
13.20	1800.54	42.46	0.0000
13.40	1750.27	25.03	0.0000

Figure 4-6. Typical Sample-Level Output Report

TIME OF FIRE	RANGE	MISS DISTANCE	PSS
5.20	3194.46	161.26	0.0000
5.40	3166.85	85.56	0.0000
5.60	3139.37	7.33	0.0028
5.80	3120.43	14.24	0.0017
6.00	3096.43	81.93	0.0000
E(KILLING HITS) = 0.04		P(KILL) =0.042	
6.20	3077.07	73.16	0.0000
6.40	3041.98	56.12	0.0000
6.60	3006.37	61.04	0.0000
6.80	2975.51	5.95	0.0032
7.00	2944.47	1.02	0.0037
E(KILLING HITS) = 0.07		P(KILL) =0.064	
7.20	2912.66	9.66	0.0027
7.40	2882.86	58.91	0.0000
7.60	2843.76	81.15	0.0000
7.80	2796.80	67.72	0.0000
8.00	2745.94	13.57	0.0019
E(KILLING HITS) = 0.04		P(KILL) =0.043	
8.20	2700.67	65.05	0.0000
8.40	2663.25	34.22	0.0000
8.60	2627.99	18.33	0.0009
8.80	2589.35	53.31	0.0000
9.00	2557.00	42.22	0.0000
E(KILLING HITS) = 0.01		P(KILL) =0.009	
9.20	2543.27	7.01	0.0039
9.40	2527.33	40.92	0.0000
9.60	2502.65	70.19	0.0000
9.80	2461.33	50.94	0.0000
10.00	2417.23	13.26	0.0021
E(KILLING HITS) = 0.06		P(KILL) =0.056	
10.20	2385.65	23.76	0.0002
10.40	2347.83	2.02	0.0057
10.60	2306.35	22.50	0.0003
10.80	2267.00	27.14	0.0001
11.00	2227.77	26.94	0.0001
E(KILLING HITS) = 0.06		P(KILL) =0.059	
11.20	2195.46	15.37	0.0015
11.40	2165.4[illegible]	18.53	0.0006
11.60	2121.35	28.65	0.0000
11.80	2093.7[illegible]	43.47	0.0000
12.00	2055.44	36.35	0.0000
E(KILLING HITS) = 0.02		P(KILL) =0.019	

Figure 4-7 Typical Sample and Burst-Level Output Report

SECTION 5
COMMUNICATION POOL

The parameter values stored in the communication pool can be viewed as three data files and a switch file. The data files, which include SENSED POSITION, PREDICTED POSITION, and SHELL POSITION, are used to organize related data and to restrict access to those modules which have a 'right' to access the data in question. The switch file is a collection of program controlled parameters which are used to alter the logical flow of the model and/or the system under study.

5.1 SENSED POSITION FILE

The content of the SENSED POSITION File is presented in Table V-1. Entries in this file are computed and stored by the Sensor Data Acquisition Event. They may be accessed only by the Prediction Algorithm Module, the Prediction Event, or a report generator.

5.2 PREDICTED POSITION FILE

The content of the PREDICTED POSITION File is presented in Table V-2. Entries in this table are computed and stored by the Prediction Event. They may be accessed by the Gun Motion and Firing Event or a report generator.

5.3 SHELL POSITION FILE

The SHELL POSITION File is filled by the Gun Motion and Firing Event and accessed by the summary report generator. The content of this file is variable since there exists several versions of the summary report generator with corresponding modified versions of the Gun Motion and Firing Event. There are, however, some general principles that can be stated. The file will always contain one entry for each occurrence of the Gun Motion and Firing Event. The entries will be ordered by time of occurrence. Entries for which no firing occurred, due to the aircraft being out of range, will be indicated by a negative sign on at least one of the time parameters in the entry. Typical output reports were presented in Section 4. Examples of entries in the file which have been implemented to generate these reports include the following:

a. Time of Fire.

   Time of Closest Approach.

   X of the shell at Closest Approach.

   Y of the shell at Closest Approach.

   Z of the shell at Closest Approach.

b. Time of Fire.

   Time of Closest Approach.

   Error in X at Closest Approach.

   Error in Y at Closest Approach.

   Error in Z at Closest Approach.

c. Time of Fire.

   Range at Closest Approach.

   Miss Distance at Closest Approach.

5.4 SWITCH FILE

There are five switches in the SWITCH File. Their function is to communicate the existence of situations which require other than normal or main-flow type computations. The five switches and their functions are:

a. *ION*. ION indicates the status of the PREDICTION Event. If the PREDICTION Event is currently scheduled, it has value one. If it is not scheduled, ION is zero. The value of ION is both set and used in the DATA ACQUISITION Event. It is, however, set and used on different entries.

b. *REGEN-TRACK*. This switch indicates whether or not regeneration should be used in the tracking computation. Its value is set by the Regenerative Tracking Switch Event and used by the Sensor Data Acquisition Event.

c. *REGEN-GUN*. This is an 'on-off' switch to control regenerative gun tracking. Its value is set at system initialization time, and remains constant throughout a simulation run. It is used in the Gun Motion and Firing Module.

d. *DKP-FLAG*. This is an 'on-off' switch for the Defense of a Known Point method of prediction. Its value is set at system initialization time and remains constant throughout a simulation run. The switch value determines the logical flow within the Prediction Event Module.

e. *JON*. JON indicates whether or not the point-to-point interpolation scheme should be used in the Prediction Algorithm Module. Its value is set by the Prediction Event Module. This switch will be 'off' unless linear bias is being used.

Table V-1. Content of the SENSED POSITION File

Parameter		Sample Time									
		t	t-Δt	t-2Δt	t-3Δt	t-4Δt	t-5Δt	t-6Δt	t-7Δt	t-8Δt	t-9Δt
Sensed Position	X	X	X	X	X	X	X	X	X	X	X
	Y	X	X	X	X	X	X	X	X	X	X
	Z	X	X	X	X	X	X	X	X	X	X
Actual Rates	$\dot{X}$	X	X	X	X	X	X	X	X	X	X
	$\dot{Y}$	X	X	X	X	X	X	X	X	X	X
	$\dot{Z}$	X	X	X	X	X	X	X	X	X	X
Smoothed Values	X										
	Y		t								
	Z										
	$\dot{X}$										
	$\dot{Y}$		$t - T_s/2$								
	$\dot{Z}$										
	$\ddot{X}$										
	$\ddot{Y}$		$t - T_s/2$								
	$\ddot{Z}$										

NOTE:

t = current clock time

Δt = sample rate

T_s = smoothing time

00834-501

Table V-2. Contents of the PREDICTED POSITION File

Predicted Parameter	Time of Computation			
	t	t-Δt	t-2Δt	t-3Δt
Range	X	X	X	X
Azimuth	X	X	X	X
Elevation	X	X	X	X
Range Rate	X			
Azimuth Rate	X			
Elevation Rate	X			
Time of Closest Approach	X			

NOTE:

t = current clock time

Δt = sample rate

00834-502

APPENDIX A
SPECIAL TRACKER MODIFICATION

During the course of the study, it became apparent that the tracking aim-wander error equation should be expanded to include the capability to model systematic errors and/or certain unique characteristics of the performance of human trackers. The form of the desired equation was known but due to a lack of funds it could not be integrated into the model. A temporary 'patch' was made which enabled the model to perform the desired computation for a fixed set of parameter values. A recompilation of the model was required for each parameter change.

The full equations for aim wander and/or systematic errors used were:

$$E_D = U_D + X_D \sigma_D \tag{A.1}$$

$$E_A = U_A + X_A [\sigma A + \lambda(Lag_A - Reg_A) \cdot D \cos(E)] \tag{A.2}$$

$$E_E = U_E + X_E [\sigma E + \lambda(Lag_E - Reg_E) \cdot D] \tag{A.3}$$

Where:

U_D = systematic error in range

U_A = systematic error in azimuth

U_E = systematic error in elevation

X_D = Normal Random variable, mean = 0, variance = 1.

X_A = Normal Random variable, mean = 0, variance = 1.

X_E = Normal Random variable, mean = 0, variance = 1.

λ = a parameter which expresses the degree of dependence of aim wander error on coordinate lag.

Lag_A = lag in azimuth

Lag_E = lag in elevation

Reg_A = regenerated azimuth rate

Reg_E = regenerated elevation rate.

Full integration of the equation into the model would enable further study in the area of human tracking. It would also open up the possibility of studying the effect of the transition from one tracking mode to another during an equipment malfunction. A failure could be modeled by a special event which, on occurrence, would modify the values of appropriate parameters in the error equation. Thus, a change in value from zero to a value appropriate to manual tracking for U_R, U_A, U_E, and λ would simulate a transition from radar to manual tracking.